Project: Model Predictive Control

Model Implementation

The model is implemented according to the course hints:

1. The main goal of the Model Predictive Control is to find a sequence of actions - actuators (steer angle, acceleration) that minimize a cost function under constraints. After the optimal sequence is found, the first pair of actions from the sequence (steer angle and acceleration) is executed and the new state is estimated according to the measurements. After that new desired trajectory and new optimal control sequence are calculated and so on.
2. The cost function includes the following terms:
    a. Squared difference between the state (cte, epsi, v) of the estimated model and the reference state. This is the most important criteria especially in the case of cte and epsi state values (ideal value is 0 –no difference from the actual vehicle position and heading to the desired position and heading). That's why it has the highest weight: 1500. The value of 1500 was estimated by several trials in which the higher values provides a little higher accuracy but lower speed and lower values provide less accuracy (if set below 1000, the car drives with higher speed but drops out of the road. The value of 1500 seems to provide an optimal trade-off between speed and driving accuracy. The weight of the speed component has much lower value (1) since it is much less important to drive at the high speed than to drive on the correct trajectory.
    b. The second component of the cost function minimizes the use of actuators i.e. take into account squared actuator values steer angle and acceleration.) Again this criterion is much less important than staying on the right trajectory, that's why this component has also much lower weight (5).
    c. Finally, the final component of the cost function is the difference between sequential actuations. The weights for the component are estimated by several trials and final values of 100 for steering angle and 10 for accelerations seem to be a good compromise between the driving accuracy and speed.

    The cost function is implemented in lines 78 – 110 of MPC.cpp.

3. Then we define constraints. Among the constraints the most important is how the state evaluates over time by using the model. This is provided by the following equations from the course material (Global Kinematic Model):

$$x_{t+1} = x_t + v_t * cos(\psi_t) * dt$$

$$y_{t+1} = y_t + v_t * sin(\psi_t) * dt$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{L_f} * \delta * dt$$

$$v_{t+1} = v_t + a_t * dt$$

These equations are implemented in lines 213 – 223 of MPC.cpp.

Timestep Length and Elapsed Duration Optimization

Timestep Length (N) and Elapsed Duration (dt) are model parameters that are optimized in order to get an optimal trade-off between driving accuracy, the speed of the vehicle and the processing time. In general increasing N would increase the driving accuracy but also increase the processing time. Also decreasing dt increases the speed of the vehicle but reduces driving accuracy.

In following table several values of N and dt are presented together with observed results. As performance criteria we used the driving accuracy (how well vehicle follows the road) and minimal speed achieved usually in the curves. The accuracy and speed were estimated qualitatively i.e. accuracy is "good" if vehicles drives closer to the middle of the road (also in curves) or "fair" if it drives near the road boarder or "bad" if it drops out of the road. Also overall speed was classified as high, medium and low.

N	dt	Speed	Driving accuracy
10	0.025	high	fair
10	**0.035**	**high**	**good**
10	0.05	medium	good
10	0.1	medium	good
20	0.01	high	bad
20	0.1	medium	good

Finally, the values of N = 10 and dt = 0.035 were selected as a good compromise between driving accuracy, driving speed and processing time.

Latency

Since the actions and thus the new state become effective for the latency duration later, we can incorporate latency by updating the new state from the old state not for the time dt but for the later time: dt + latency. Latency is taken into account in the state equitation i.e. x_{t+1}, y_{t+1}, $\psi_{\tau+1}$ and v_{t+1} (see above) by using the terms (dt + latency) instead dt.

This is implemented in lines 213 – 223 of MPC.cpp with the latency value of 0.1 i.e. 100 ms.